Optimal Filtering

Problem:

- How to estimate one signal from another.
- In many applications desired signal is not observable directly (convolved with another, distorted by noise).

ex: (1) Information signal transmitted over channel gets corrupted with noise.

(2) Image recorded by system is subject to distortions.

(3) Directional antenna array is vulnerable to string jammers in other directions due to sidelobe leakage, etc.

In this course:

$\rightarrow$ Emphasis on least square techniques to estimate/recover signal (i.e., case $\|\cdot\|_2$).

$\downarrow$

not the only way to solve problems ($\|\cdot\|_p$ could be used).

Look at:

- Orthogonality Principle.
- Wiener filtering (FIR, IIR).

Estimation of Signals

$\underline{x} \longrightarrow \boxed{\text{optimal processor}} \longrightarrow \hat{y} =$ *best* estimate of y (where y is related to $\underline{x}$)

Possible procedure: Mean square estimation

$$\longrightarrow \text{minimize:} \quad \xi = E\left\{ \left| y - \hat{y}(\underline{x}) \right|^2 \right\}$$

may not be linear in $\underline{x}$

$$\longrightarrow \boxed{\hat{y}(\underline{x}) = E[y \mid \underline{x}]}$$

conditional mean

Proof:
$$\xi = E\left\{ \left| y - \hat{y}(\underline{x}) \right|^2 \right\}$$
$$= E\left\{ \underbrace{\left(y - \hat{y}(\underline{x})\right)\left(y - \hat{y}(\underline{x})\right)^T} \right\}$$

Define $L\left(y, \hat{y}(x)\right)$: loss function

$$\Rightarrow \hat{y}(x) = E[y \mid \underline{x}]$$

- **Generally** $\hat{y}(\underline{x})$ **is a non-linear function of** $\underline{\boldsymbol{x}}$

  [exception when $\underline{x}$ and y are jointly normal Gauss-Markov theorem]

- **Complicated to solve**, sometimes no closed-form solution.

- **Restriction to Linear Mean Square Estimator (LMS)**

  estimator of y is **forced** to be a linear function of measurements $\underline{x}$:

  $\hat{y} = \underline{a}^H \underline{x}$

  will produce the **Wiener-Hopf equations**

- Derivation of W-H eqs. (greatly!) simplified by use of **orthognality principle**.

Note: LMS error is **never** smaller than MS error (why?)

Orthogonality Principle

Use LMS Criterion: estimate y by $\hat{y} = \underline{a}^H \underline{x}$
where weights $\{a_i\}$ minimize
MS error:

$$\sigma_\varepsilon^2 = E\left\{\left|y - \hat{y}(\underline{x})\right|^2\right\}$$

Th: Let error $\varepsilon = y - \hat{y}$

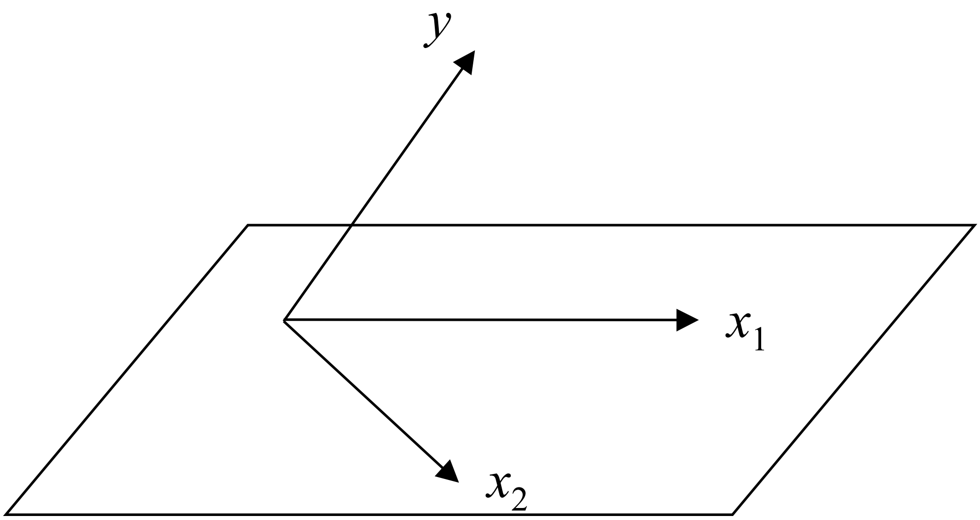
$\underline{a}$ minimizes the MSE σ_ε^2 if $\underline{a}$ is chosen such that

$$E\left\{\varepsilon x_i^*\right\} = E\left\{x_i^* \varepsilon\right\} = 0, \quad \forall_i = 1, \cdots, N$$

(i.e., error ε is orthogonal to observations $\left\{x_i\right\}_{i=1}^N$).

Corollary: minimum MSE obtained:

$$\sigma_{\varepsilon \min}^2 = E\left\{y\varepsilon^*\right\}$$



y
x_1
x_2

Proof: $\varepsilon = y - \underline{a}^H \underline{x} = y - \left(\underline{a} + \underline{\hat{a}} - \underline{\hat{a}}\right)^H \underline{x}$

where $\underline{\hat{a}}$ is the weight vector defined so that the orthogonality principle holds. Resulting error is called $\hat{\varepsilon} = y - \underline{\hat{a}}^H \underline{x}$

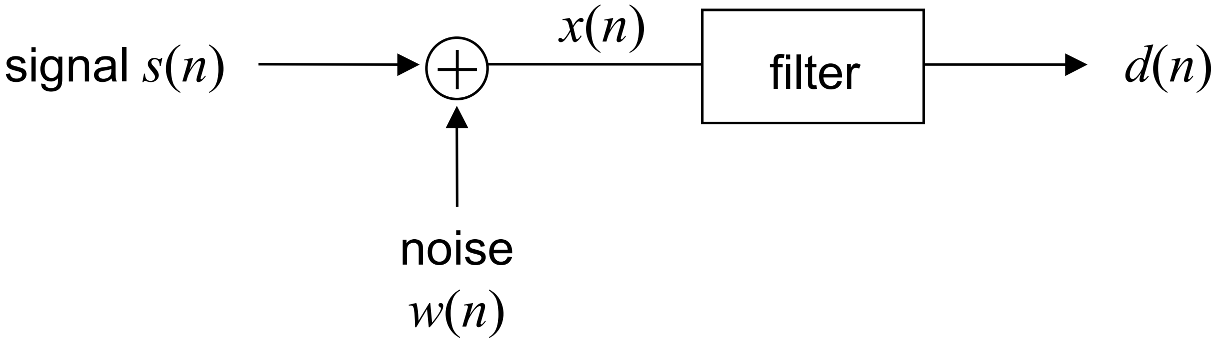
Optimal Filtering - Wiener Filtering

Problem: estimate signal $\hat{d}(n)$ from observations $x(n)$ [noisy version of $d(n)$]

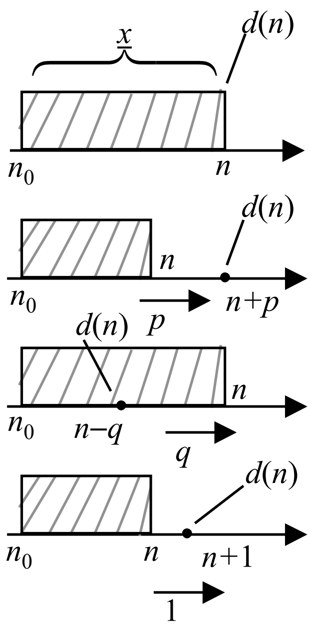
Different Wiener filtering problems:

- smoothing
- filtering
- prediction

difference between above Wiener problems comes from which of the **available** observations are taken into account to compute the filter coefficients.



Typical Wiener Filtering Problems



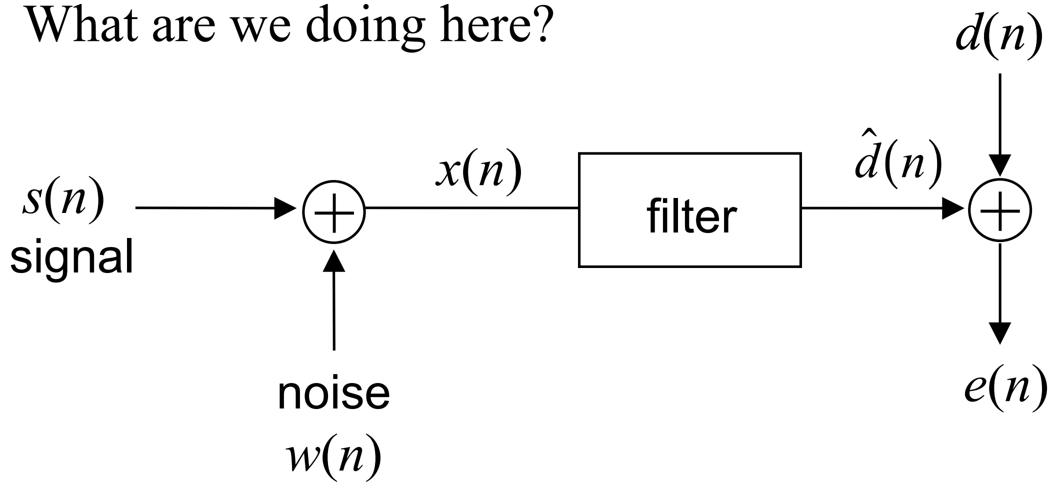
Problem	Form of Observations	Desired Signal
Filtering of signal in noise	$x[n] = s[n]+\eta[n]$	$d[n] = s[n]$
Prediction of signal in noise	$x[n] = s[n]+\eta[n]$	$d[n] = s[n+p]$; $p > 0$
Smoothing of signal in noise	$x[n] = s[n]+\eta[n]$	$d[n] = s[n-q]$; $q > 0$
Linear prediction	$x[n] = s[n-1]$	$d[n] = s[n]$
General nonlinear problem	$x[n] = G(s[n], \eta[n])$	$d[n] = s[n]$

Wiener Filtering

FIR Case

(1) Wiener FIR case

- Filter criterion used: minimization of mean square error between d(n) and $\hat{d}(n)$.
- What are we doing here?



We want to design a filter (in the generic sense can be: filter, smoother, predictor) so that:

$$\hat{d}(n) = \sum_{k=0}^{P-1} h^*(k)x(n-k)$$

How $d(n)$ is defined specifies the operation done:

- filtering:
- predicting:
- smoothing:

Wiener filter can be derived for non-stationary processes:

How to find *h*(*k*)?

Minimize the MSE: $E\left\{\left|d(n)-\hat{d}(n)\right|^2\right\}$

$$\sum_{k=0}^{P-1} h^*(k)x(n-k) = \underline{h}^H \underline{x}$$

$$\underline{h} = \left[h(0), \text{——} \ h(P-1)\right]^T$$

$$\underline{x} = \left[x(n), \text{——} \ x(n-P+1)\right]^T$$

Wiener filter is a linear filter $\Rightarrow$ orthogonality principle applies

$$\Rightarrow E\left\{x(n-i)\varepsilon^*(n)\right\} = 0 \qquad \forall i =$$

$$E\left\{x(n-i)\left[d(n)-\sum_{k=0}^{P-1} h^*(k)x(n-k)\right]^*\right\} = 0 \qquad \forall i =$$

$\Rightarrow$

Minimum MSE(MMSE)

obtained when $\underline{h} = \underline{h}_{opt}$ is the optimum weight vector

Matrix formulation of Wiener equations:

- Recall: $\hat{d}(n) =$

  where –

  –

  –

- Orthogonality principle:

$$E\left[\varepsilon(n)x_i^*\right] = E\left[x_i\varepsilon^*(n)\right] = 0$$
$$\forall i = 0, \rule{2cm}{0.4pt} P-1$$

where $\{x_i\}$ represent the set of observations used to compute the filter output.

In this case $\{x_i\}=$

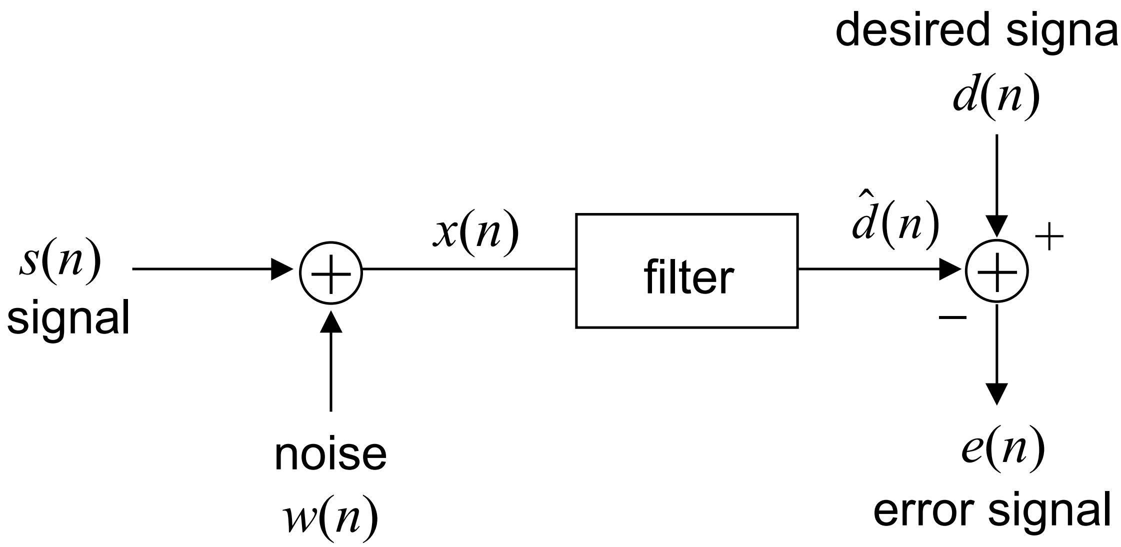
Minimum MSE obtained for optimum weights:

Recall orthogonality principle says:

$$\sigma^2_{\varepsilon\min} = E\left[d(n)\varepsilon^*(n)\right]$$

$$=$$

Summary: Wiener Filter Equations



- Wiener filter is a filter such that:

$$\hat{d}(n) = \sum_{k=0}^{P-1} h(k)^{*} x(n-k)$$

<u>such that</u>: $\sigma_e^2 = E\left[\left|d(n) - \hat{d}(n)\right|^2\right]$ <u>minimum</u>

- How $d(n)$ is defined **specifies** the specific type of Wiener filter designed:

  filtering:

  smoothing:

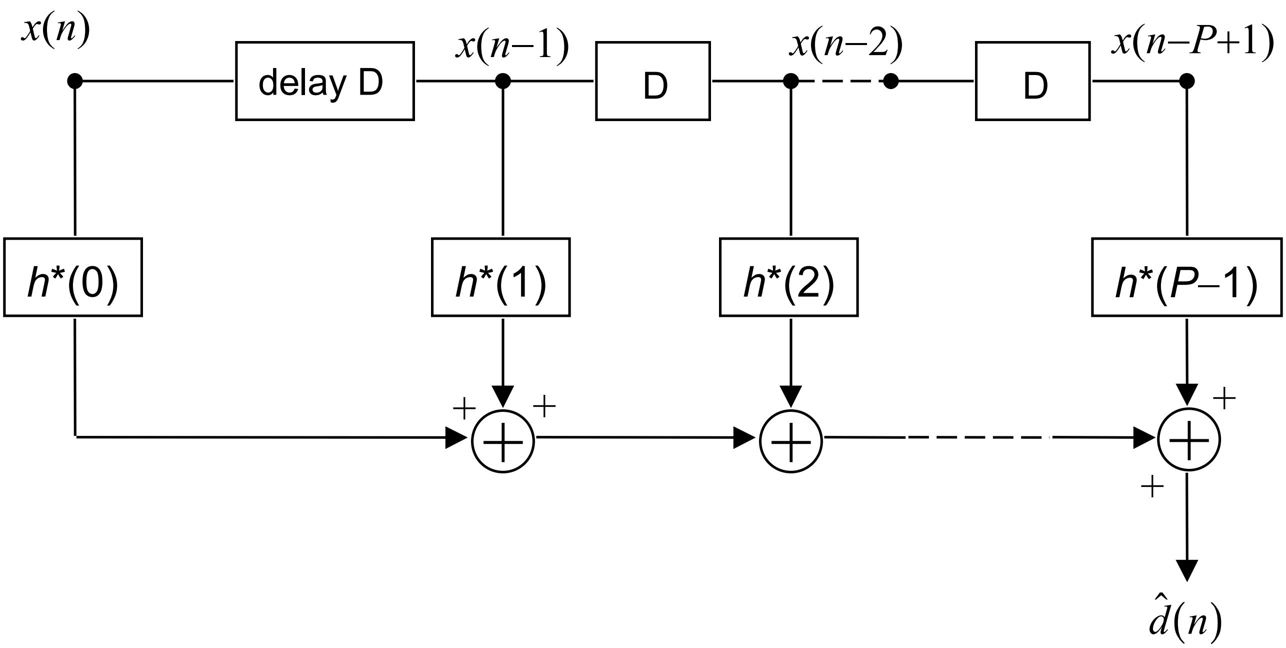
  predicting:

- $\rightarrow$ w-H eqs.: $h_{\text{opt}} =$

  MMSE: $\sigma^2_{e\min} =$

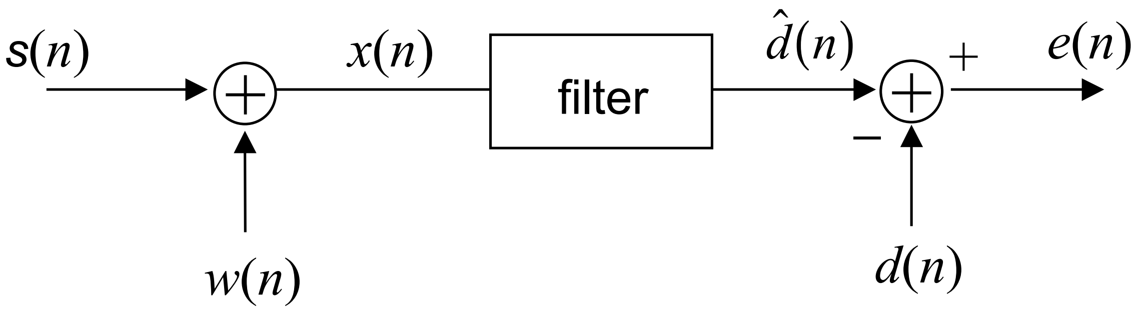
- The FIR implementation permits use of a tapped-delay-line (TDL) filter with a **finite** number of coefficients.

$$\hat{d}(n) = \sum_{\ell=0}^{P-1} h^*(\ell)x(n-\ell)$$



Application to Wiener filter ($d(n) = s(n)$)

example 1: assume x(n) is defined by



$s(n)$, $w(n)$ uncorrelated

$w(n)$ white noise, zero mean $\quad R_w(n) - 2\delta(n)$

$s(n)$ $\quad R_s(n) = 2\,(0.8)^{|n|}$

example 2: $s(n)$, $w(n)$ uncorrelated

$w(n)$ noise with $R_w(n) = 2\,(0.5)^{|n|}$

$s(n)$ signal with $R_s(n) = 2\,(0.8)^{|n|}$

Application of Wiener filter to one-step linear prediction

- tracking of moving series
- forecasting of system behavior
- data compression
- telephone transmission

- W-H eqs. $\longrightarrow$

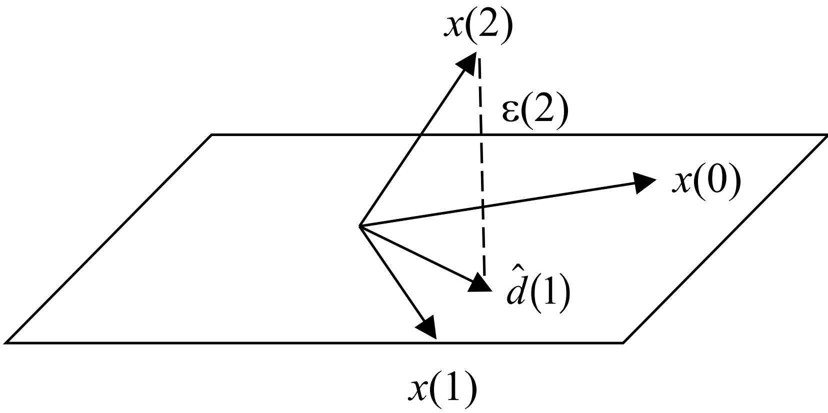
$$\underline{h}_{\text{opt}} = R_x^{-1} \underline{r}_{dx}^{*}$$

where $$\hat{d}(n) = \sum_{\ell=0}^{P-1} h^{*}(\ell)\, x(n-\ell)$$

$$d(n) = \;?$$

- Geometric interpretation: Assume

$P = 1$ (filter of length 2)
$n = 1$



$\hat{d}(1) =$

$\varepsilon(2) =$

ε(2) is the error between true value $x(2)$ and predicted value for $x(2)$ based on $x(1)$ and $x(0)$

⟶ represents the new information in $x(2)$ which is not contained in $x(0)$ and $x(1)$

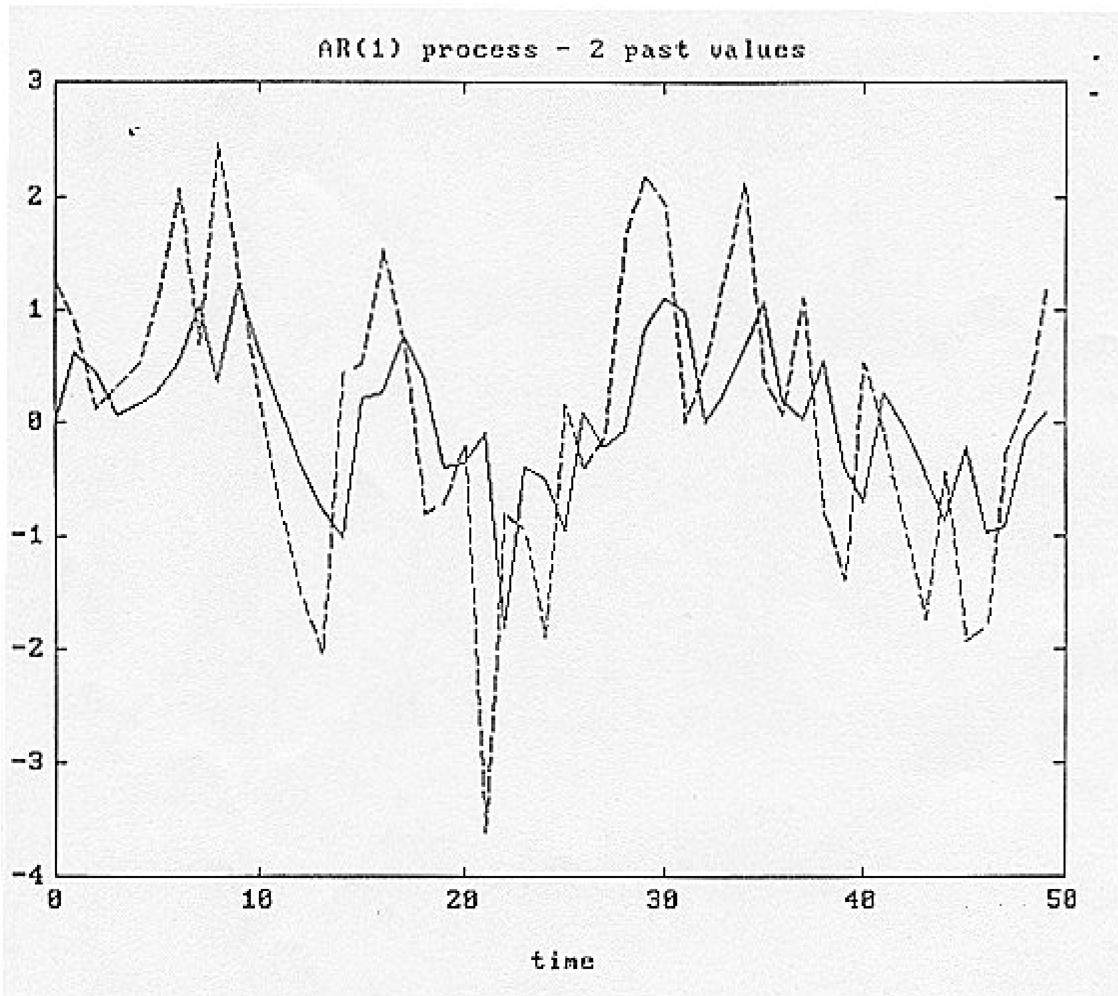
⟶ ε(n) is called the **innovation process** corresponding to $x(n)$

ex: $x(n) = a\, x(n-1) + n(n) \quad |a| < 1$

$n(n)$ is white noise

AR (1) process
predictor

$a_p = 0.5$



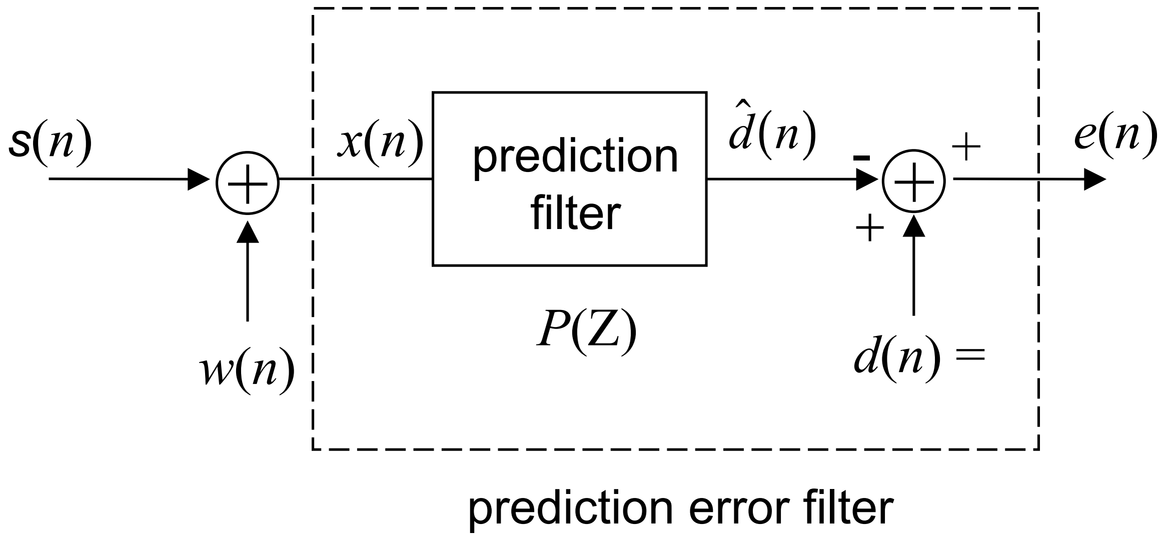
seed = 1024

$$\hat{x}(n) = +a_1 x(n-1) + a_2 x(n-2)$$

$$a = \begin{bmatrix} +0.5 \\ 0 \end{bmatrix}$$

Prediction Error Filter (PEF) Definition

Recall:



- <u>Prediction error filter transfer function</u>

Example 1:

$s(n)$ = process with $R_s(n) = 2(0.8)^{|n|}$

$w(n)$ = white noise, zero mean $R_w(n) = 2\delta(n)$

$s(n)$, $w(n)$ uncorrelated

Design the 2-step ahead predictor of length 3.
Compute MMSE.

Example 2:

$s(n)$ = process with $R_s(n) = 2(0.8)^{|n|}$

$w(n)$ = white noise, zero mean $R_w(n) = 2(0.5)^{|n|}$

$s(n)$, $w(n)$ uncorrelated

- Design the 1-step ahead predictor by length $\begin{cases} 2 \\ 3 \end{cases}$
- Design 1-step back smoother of length $\begin{cases} 2 \\ 3 \end{cases}$

Review of Spectral Factorization Concepts

(to be used for Wiener filtering -- IIR implementation)

Pb: Given the PSD (power spectral density) $s_n(\omega)$, find a minimum phase function $H(j\omega)$ so that

$$|H(j\omega)|^2 = S_x(\omega)$$

↓

Pb has a solution of $S_x(\omega)$ satisfies: Paley-Wiener conditions

$$\int_{-\infty}^{+\infty} \frac{\left|In\left(S_x(\omega)\right)\right|}{1+\omega^2} d\omega < \infty$$

↓

Condition not satisfied if $S_x(\omega)$ has discrete lines.

↓

Pb not easy; restrict to rational spectrum $S_x(\omega)$.

For discrete systems:

Pb has solution $S_x(z) = H(z).\ \ H^*\left(1/z^*\right)$ if it satisfies Paley-Wiener condition:

$$\int_{-\pi}^{\pi} In\left[S_x\left(e^{j\omega}\right)\right]d\omega < \infty \qquad \text{for} \quad z = e^{j\omega}$$

Pb not easy: restrict to rational spectrum.

Def: $x(n)$ has a rational spectrum if:

$$S_x(z) = \frac{A(z)}{B(z)}$$

$$= k_0 \underbrace{\frac{N(z)}{D(z)}} \quad \underbrace{\frac{N^*(1/z^*)}{D^*(1/z^*)}}; \qquad k_0 > 0$$

normalizing constraint

"inside" group
all poles and zeros
inside u.c. $|z| < 1$
$H(z) = H_{\min}(z)$
<u>minimimum phase</u>
<u>component</u>

"outside" group
all poles and zeros
outside u.c. $|z| > 1$
$H^*(1/z^*) = H_{\max}(z)$
<u>maximum phase</u>
<u>component</u>

Note: "minimum phase" doesn't mean $H_{\min}(z)$ has minimum phase

→ $H_{\min}(z)$ <u>is actually with maximum phase</u> over all filters with same magnitude response.

↓

$H_{\min}(z)$ <u>has minimum phase lag</u> = $-\underline{\angle H(e^{j\omega})}$

(see Chapter 5 - textbook).

Note: Minimum phase system is a causal system with a minimum phase transfer function.

Brief review of discret process properties

- **Property:** for real RPs $x(n)$ poles and zeros occur in groups at:

$$z_o, z_0^*, 1/z_o, 1/z_0^*$$

- **Proof:**

$$R_x(k) = R_x^*(-k) \Rightarrow$$

Ex: $S_x(z) = \dfrac{5 - 2z - 2z^{-1}}{10 - 3z - 3z^{-1}}$

Find minimim phase and maximum phase components.

Wiener filter -- IIR implementation

- IIR sometimes easier to implement
- Requires solving for a closed form solution for filter weights
- Recall FIR implementation gives the following

$$\hat{d}(n) = \sum_{k=0}^{P-1} h^*(k) x(n-k)$$

- IIR implementation: $\boxed{\text{Change in notations}}$

- Again we want to find $\underline{h}$ so that:

$$\sigma_\varepsilon^2 = E\left[\left|d(n) - \hat{d}(n)\right|^2\right] \text{ is minimum}$$

Use orthogonality principle

$$E\left\{x(n-i)\varepsilon^{*}(n)\right\} = E\left\{\varepsilon(n)x^{*}(n-i)\right\} = 0 \qquad \forall\, i = 0,\cdots,\infty$$

Minimum mean square error

$$\sigma^2_{\varepsilon_{\min}} =$$

$$=$$

$$=$$

How to compute *h*?

$\longrightarrow$ use “Prewhitening Approach”

- **Pb:** solve

$$R^*_{dx}(i) = \sum_{\ell=0}^{\infty} h^*(\ell) R_x(\ell - i) \quad i = 0, \cdots, +\infty \qquad (1)$$

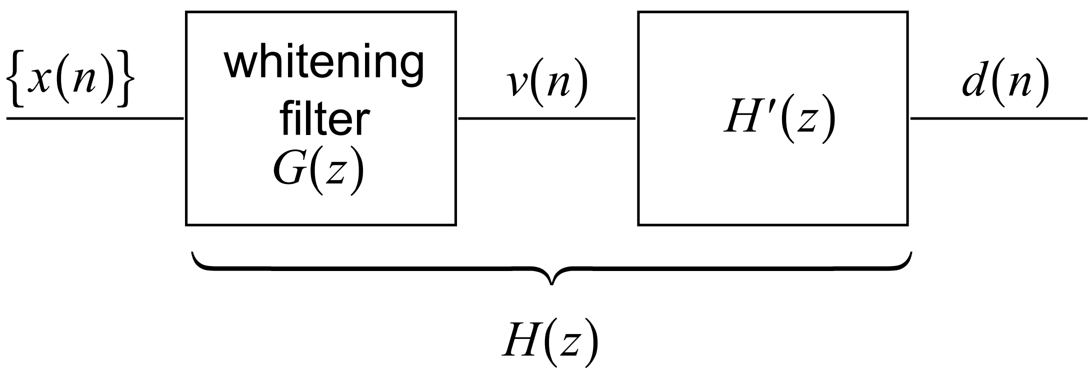
- **Note:** (1) can’t use Z-transforms (why?)

  (2) if $x(n)$ = white noise $\Rightarrow R_x(n) =$

  $\longrightarrow$ then (1) $\Rightarrow$

Solution of IIR Causal Wiener Pb

Idea: decompose filter into 2 filters so that $v(n)$ is white



$\Rightarrow$ **Pb now** is to find $H'(z)\left[h'(n)\right]$ so that $H(z)$ is the optimum Wiener filter

6-Step Process:

(1) Use Wiener-Hopf equation on $v(n)$, d(n) to find $h'(n)$

$$\sum_{\ell=0}^{\infty} h'^{*}(\ell) R_v(\ell - i) = R_{dv}^{*}(i), \quad \forall\ i = 0, \cdots, +\infty$$

assuming $v(n)$ is white

$$R_v(n) = \sigma_v^2 \delta(n)$$

$$\Downarrow$$

$$h'(i) =$$

(2) Compute $R_{dv}(i)$

Recall $v(n) =$

$\Rightarrow \quad R_{dv}(i) =$

(3) Compute $g(n)$ or $G(z)$

[whitening filter characteristics]

(4) Compute *H′* (*z*)

Recall

$$h'(i) = \begin{cases} \dfrac{R_{dv}(i)}{\sigma_v^2} & \forall\, i = 0, \cdots, \infty \\ 0 & \text{otherwise} \end{cases}$$

$$\Rightarrow \quad H'(z) =$$

(5) Compute S_{dv} (*z*)

from (2) we know $R_{dv}(i) = \sum_k g^*(k) R_{dx}(i+k)$

$$=$$

$$\Rightarrow S_{dv}(z) =$$

$$\Rightarrow \quad H'(z) =$$

(6) Overall filter $H(z) = G(z)\,H'z$

$H\,(z) =$

Summary of steps involved in the computation of *H*(*z*)

- Given $S_x(\omega)$, **form** $S_x(z)$ by setting $e^{j\omega} = z$ if $S_x(z)$ is not given directly.
- **Factor** $S_x(z)$ as $\sigma^2 H_{ca}(z)$. $H_{ca}^*(1/z^*)$
- **Assign** all poles and zeros of $S_x(z)$ inside unit circle to $H_{ca}(z)$.
- **Assign** ———— outside ———— $H_{ca}^*(1/z^*)$
- **Remember** to define $H_{ca}(z)$ so that $H_{ca}(+\infty) = 1$
- **Form** $S_{dx}(z)$ and $S_{dx}(z)/H_{ca}^*(1/z^*)$
- **Identify** causal part of $S_{dx}(z)/H_{ca}^*(1/z^*)$
- **Form** $H(z)$

How to identify the causal and stable part of $X(z)$

(1) Stability

Recall:

- a discrete system S is stable if and only if its impulse response is absolutely summable.

  i.e., $\sum_n |h(n)| < +\infty$

- $H(z) = z\,[\,h(n)\,]$ transfer function

  (when $z = e^{j\omega}$ $H(e^{j\omega})$: frequency response)

- Region of Convergence (ROC) of $H(z)$ is defined by values of z for which:

$\Rightarrow$ System is stable (i.e., has a stable frequency response) if its ROC includes the unit circle (z so that $|z| = 1$).

(2) Causality:

Assume a sequence $x(n) = 0,\ n < n_1$ (right-sided sequence)

$$X(z) = \sum_{n=n_1}^{\infty} X(n) z^{-n} \qquad n_1 = 0$$

Property: The ROC of $X(z)$ is the center of a circle.

- **Proof**

Conclusion:

A causal and stable discrete system must have a system function $H(z) = z\ [\ h(n)\ ]$ with ROC including:

- the unit circle
- entire z-plane outside the unit circle including $z = \infty$

Evaluation of MMSE for causal IIR Wiener filter

(1) Using the time-domain

$$\sigma^2_{\varepsilon_{\min}} = R_d(0) - \sum_{k=0}^{\infty} h(k) R_{dx}(k)$$

→ too complicated; requires $R_{dx}(k) \quad k = 0, \to \infty$

$h(k) \quad k = 0, \to +\infty$

(2) Using the frequency domain

Recall

$$\sigma^2_{\varepsilon_{\min}} = E\left[d(h)\varepsilon^*(n)\right]$$

$$\Rightarrow \quad R_{d\varepsilon}(\ell) = z^{-1}\left[S_{d\varepsilon}(z)\right] = \frac{1}{2\pi j}\oint S_{d\varepsilon}(z) z^{-1} dz$$

$$\Rightarrow \quad R_{d\varepsilon}(0) = \frac{1}{2\pi j}\oint S_{d\varepsilon}(z) z^{-1} dz$$

$$\Rightarrow \quad \boxed{\sigma^2_{\varepsilon_{\min}} = \sum_k \mathrm{Res}\left[S_{d\varepsilon}(z) z^{-1}, z_k\right]}$$

where z_k is a pole of $S_{d\varepsilon}(z)\, z^{-1}$ contained inside the unit circle.

Note: Why is $\sigma^2_{\varepsilon_{\min}}$ evaluated for poles inside the unit circle?

Only because $S_{d\varepsilon}(z)$ is two-sided

$\Rightarrow S_{d\varepsilon}(z)$ converges in the region

$\rho < |z| < 1/\rho.$

we have to pick a path which is inside the ROC

$\longrightarrow$ pick path = unit circle

Solution of IIR non-causal Wiener Pb

for off-line applications only

$H_{nc}(z)$ defined by:

$$\begin{cases} \displaystyle\sum_{k=-\infty}^{+\infty} h_{nc}^*(k) R_x(k-n) = R_{dx}^*(n) \quad \forall\ n & \circledast \\ \displaystyle\sigma_{\varepsilon_{\min}}^2 = R_d(0) - \sum_{k=-\infty}^{+\infty} h_{nc}^*(k) R_{dx}(k) & \end{cases}$$

$\circledast \Rightarrow$

Comments on IIR non-causal Wiener filter results

The results are of theoretical interest for the following reasons:

1. The mean-square error ε_u obtained by using an unrealizable Wiener filter provides a lower bound on the mean-square error that is attainable by any realizable linear filter. The mean-square error, ε_u, is therefore referred to as the *irreducible* (or *infinite delay*) *error*.

2. The use of an unrealizable Wiener filter has the same effect as a realizable Wiener filter obtained by allowing an infinite delay in the desired response. This result may be justified intuitively as follows. In the case of an unrealizable Wiener filter, the use of the entire past and future values of the input signal is permitted to produce an estimate of the desired response at the present time. We may build a realizable filter whose performance approaches that of the unrealizable Wiener filter by waiting until more of the future values of the input signal come in to produce an estimate of the desired response at a later time. In this way we can produce a mean-square error that is arbitrarily close to the irreducible error ε_u by increasing the delay in the desired response.